CLAIMS

We claim:

1. A method for identifying erroneous transactions that occur during processor architecture verification testing, the method comprising:

monitoring an interface;

determining information related to termination of a test case; and

after the test case has terminated, identifying an incomplete transaction that should have completed prior to termination of the test case.

2. The method of claim 1, wherein monitoring an interface comprises monitoring a point-to-point (P2P) link network.

3. The method of claim 1, wherein monitoring an interface comprises monitoring a point-to-point (P2P) link network of a register transfer language (RTL) simulator.

4. The method of claim 1, wherein determining information related to termination of a test case comprises detecting a break signal asserted on the interface.

5. The method of claim 1, wherein determining information related to termination of a test case comprises receiving an indication that a test model has stopped processing.

6. The method of claim 1, wherein identifying an incomplete transaction comprises consulting a pending transactions list.

7. The method of claim 6, wherein identifying an incomplete transaction further comprises filtering out pending transactions of a type that does not likely indicate an error.

8. The method of claim 7, wherein filtering out pending transactions comprises disregarding at least one of transactions having a start time on or after the time at which a break signal was asserted and transactions that occur as a result of a break command being issued.

9. The method of claim 7, wherein identifying an incomplete transaction further comprises flagging all pending transactions other than those that were filtered out.

10. The method of claim 9, wherein flagging all other transactions comprises alerting a user as to the existence of the pending transactions and providing information to the user that can be used to determine the reason why those transactions did not complete.

11. The method of claim 10, wherein providing information comprises at least one of providing all completed packets associated with the transactions and providing a summary of each transaction that describes all processing associated with each transaction.

12. A system for identifying erroneous transactions, the system comprising:

means for monitoring all ports of an interface;

means for determining information related to termination of a test case; and

means for identifying an incomplete transaction that should have completed prior to termination of the test case.

13. The system of claim 12, wherein the means for monitoring comprise means for monitoring a point-to-point (P2P) link network of a register transfer language (RTL) simulator.

14. The system of claim 12, wherein the means for determining information comprise means for detecting break signals asserted on the interface.

15. The system of claim 12, wherein the means for determining information comprise means for receiving an indication that a test model has stopped processing.

16. The system of claim 12, wherein the means for identifying an incomplete transaction comprise a pending transactions list.

17. The system of claim 16, wherein the means for identifying an incomplete transaction comprise means for disregarding at least one of transactions having a start time on or after the time at which a break signal was asserted and transactions that occur as a result of a break command being issued.

18. The system of claim 12, wherein the means for identifying an incomplete transaction comprise means for flagging all pending transactions that are determined to be erroneous.

19. The system of claim 18, wherein the means for flagging comprise means for at least one of providing all completed packets associated with the transactions and providing a summary of each transaction that describes all processing associated with each transaction.

20. A virtual bus interface (VBI) stored on a computer-readable medium, the VBI comprising:

logic configured to monitor a point-to-point (P2P) interface;

logic configured to determine a time at which a break signal was asserted;

logic configured to identify transactions that are pending after the break signal was asserted; and

logic configured to determine which of the pending transactions are erroneous.

21. The VBI of claim 20, wherein the logic configured to identify transactions comprises logic configured to consult a pending transactions list.

22. The VBI of claim 20, wherein the logic configured to determine comprises logic configured to filter out pending transactions of a type that does not likely indicate an error.

23. The VBI of claim 22, wherein the logic configured to filter out pending transactions comprises logic configured to disregard at least one of transactions having a start time on or after the time at which the break signal was asserted and transactions that occur as a result of a break command being issued.

24. The VBI of claim 20, further comprising logic configured to flag erroneous transactions.

25. The VBI of claim 24, wherein the logic configured to flag comprises logic configured to alert a user as to the existence of the erroneous transactions and logic configured to provide information to the user that can be used to determine the reason why those transactions did not complete.

26. A processor architecture verification system, comprising:

a register transfer language (RTL) simulator that simulates operation of a processor and generates a first output in a first format, the RTL simulator including an interface;

a golden simulator that simulates operation of the processor and generates a second output in a second format;

a translator that translates at least one of the outputs for comparison with the other output, the translator including a virtual bus interface (VBI) that comprises logic configured to monitor a point-to-point (P2P) interface, logic configured to determine a time at which a break signal was asserted, logic configured to identify transactions that are pending after a break command was issued, and logic configured to determine which of the pending transactions are erroneous; and

a comparator that compares the first and second outputs after translation of the at least one output.

27. The system of claim 26, wherein the logic configured to identify transactions comprises logic configured to consult a pending transactions list.

28. The system of claim 26, wherein the logic configured to determine comprises logic configured to filter out pending transactions of a type that does not likely indicate an error.

29. The system of claim 26, wherein the logic configured to filter· out pending transactions comprises logic configured to disregard at least one of transactions having a start time on or after the time at which the break signal was asserted and transactions that occur as a result of a break command being issued.